GETTEMPPATH

Vulnerable to several path and buffer issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6541 bytes

<table>
<tr><td><b>Attack Category</b></td><td>• Environment Manipulation
• Path spoofing or confusion problem</td></tr>
<tr><td><b>Vulnerability Category</b></td><td>• Temporary file creation problem
• Buffer Overflow
• Privilege escalation problem</td></tr>
<tr><td><b>Software Context</b></td><td>• File Path Management</td></tr>
<tr><td><b>Location</b></td><td></td></tr>
<tr><td><b>Description</b></td><td>GetTempPath() returns the file path to the temporary directory.

"The GetTempPath function checks for the existence of environment variables in the following order and uses the first path found:
1. The path specified by the TMP environment variable.
2. The path specified by the TEMP environment variable.
3. The path specified by the USERPROFILE environment variable.
4. The Windows directory.
Note that the function does not verify that the path exists."

GetTempPath() raises two concerns.
First, a buffer overflow condition could exist if the path to the temporary directory is longer than the buffer allocated to store this information.
Second, a path to an insecure directory could be returned.

Also, Windows does not guarantee that the returned paths are valid or useable (e.g. writable) for temporary files.</td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>GetTempPath</td><td></td></tr><tr><td>GetTempPathA</td><td></td></tr></table></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td></td><td><table><tr><td>GetTempPathW</td><td></td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td>An attacker could take advantage of either weakness in the functionality of GetTempPath(). First, if Windows returns a path to which the attacker can read or write (e.g. c:\temp), he or she will be able to read or alter any data in the temporary files. This would result in a breach of confidentiality and integrity, respectively.
The second attack vector is to implement a buffer overflow attack. There is no indication that the value in any of the environment variables (TMP, TEMP) is truncated to MAX_PATH. Therefore, an attacker could specify an environment variable whose length is longer than that of the path buffer if the buffer's length is not set properly. When the program was run in this environment, the path buffer would be overflowed.</td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td><table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>This solution is always applicable.</td><td>Check to make sure that path buffer is of length MAX_PATH + 1 (to allow for a null character).</td><td>This will prevent buffer overflows.</td></tr>
<tr><td>This solution is always applicable.</td><td>When a path has been returned, check its validity and security. If the path is valid, make sure any files created have the most restrictive amount of permissions necessary. Additionally, it would be wise to make sure that any directory used as a temporary directory cannot have parent permissions flushed down</td><td>This solution will mitigate the risk of insecure or unavailable temporary files.</td></tr>
</table></td></tr>
</table>

	to overwrite the permissions of the children which would be the temporary files we are trying to protect.
Signature Details	DWORD GetTempPath(DWORD nBufferLength, LPTSTR lpBuffer);

Examples of Incorrect Code	`/* Improper sizing of the buffer */`
	`LPTSTR path_buffer [20];`
	`GetTempPath(MAX_PATH, path_buffer);`

Examples of Corrected Code	`/* Use of a properly sized buffer */`
	`LPTSTR path_buffer [MAX_SIZE + 1]; //Be sure the buffer can hold a path that is as long as the OS allows.`
	`if (GetTempPath(MAX_PATH + 1, path_buffer) == 0) //Fill the buffer up to it's length.`
	`return -1; //Handle any errors that occur.`
	`//Check to make sure the path is valid,writable, and secure.`
	`if (!CheckPermissionsAndValidityOfPath(path_buffe`
	`return -1;`

Source Reference	- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/fileio/fs/gettemppath.asp[2]
Recommended Resource	- MSDN reference for GetTempPath[3]
Discriminant Set	**Operating System**: - Windows **Language**:

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com